



# Business process improvement by means of Big Data based Decision Support Systems: a case study on Call Centers

## **Alejandro Vera-Baquero**

Universidad Carlos III de Madrid  
Av. Universidad 30, Leganes (Madrid) 28911  
Spain  
[www.shortbio.net/averabaq@gmail.com](http://www.shortbio.net/averabaq@gmail.com)

## **Owen Molloy**

National University of Ireland  
Galway  
Ireland  
[www.shortbio.net/owen.molloy@nuigalway.ie](http://www.shortbio.net/owen.molloy@nuigalway.ie)

## **Ricardo Colomo-Palacios**

Østfold University College  
B R A Veien 4, Halden 1783  
Norway  
[www.shortbio.net/ricardo.colomo-palacios@hiof.no](http://www.shortbio.net/ricardo.colomo-palacios@hiof.no)

## **Mahmoud Elbattah**

National University of Ireland  
Galway  
Ireland  
[www.shortbio.net/m.elbattah1@nuigalway.ie](http://www.shortbio.net/m.elbattah1@nuigalway.ie)

### **Abstract:**

Big Data is a rapidly evolving and maturing field which places significant data storage and processing power at our disposal. To take advantage of this power, we need to create new means of collecting and processing large volumes of data at high speed. Meanwhile, as companies and organizations, such as health services, realize the importance and value of "joined-up thinking" across supply chains and healthcare pathways, for example, this creates a demand for a new type of approach to Business Activity Monitoring and Management. This new approach requires Big Data solutions to cope with the volume and speed of transactions across global supply chains. In this paper we describe a methodology and framework to leverage Big Data and Analytics to deliver a Decision Support framework to support Business Process Improvement, using near real-time process analytics in a decision-support environment. The system supports the capture and analysis of hierarchical process data, allowing analysis to take place at different organizational and process levels. Individual business units can perform their own process monitoring. An event-correlation mechanism is built into the system, allowing the monitoring of individual process instances or paths.

### **Keywords:**

business process improvement; Big Data; Decision Support Systems; processes.

**DOI:** 10.12821/ijispm030101

**Manuscript received:** 9 September 2014

**Manuscript accepted:** 17 November 2014

## 1. Introduction

The Internet era has led to the production of increasing amounts of digital data [1]. Big Data (BD) is an emerging phenomenon [2] which has drawn huge attention from researchers in information sciences, as well as policy and decision makers in governments and enterprises [3]. There are several definitions for the term, however, the predominant one seems to be that BD relates to datasets that have become too large to handle with the traditional or given computing environment [4]. It is fair to say that the Information Technology (IT) world has been facing BD challenges for over four decades, but the definition of “big” has been changing from megabytes in the 1970s to the petabyte range today [5]. Within this phenomenon, [6] points out that BD can be seen as two different issues: big throughput and big analytics; the former includes the problems associated with storing and manipulating large amounts of data and the latter those concerned with transforming this data into knowledge. Focusing on the analytics, BD analytics is a workflow that distills Terabytes of low-value data down to, in some cases, a single bit of high-value data with the goal to see the big picture from the minutiae [7]. This new discipline requires new approaches to obtain insights from highly detailed, contextualized, and rich contents that may require complex math operations, such as machine learning or clustering [2]. This diversity of tools and techniques for BD-driven analytics systems makes the process nontrivial. In the analytics of these kind of systems several artificial intelligence technologies play a crucial role [8]–[10].

On the other hand, LaValle et al. [11] report that top-performing organizations ‘make decisions based on rigorous analysis at more than double the rate of lower performing organizations’ and that in such organizations analytic insight is being used to ‘guide both future strategies and day-to-day operations’. In sum, literature reported significant interest in the potential of ‘big data’ and ‘analytics’ to transform the competitive landscape and to improve organizational performance [12]. Examples of the use of big data can be found in several sectors, including government [13], academia [14], medicine [15], climate science [16] and agriculture [17].

One of the main tools employed in organizations are Decision Support Systems (DSS). DSS are computer technology solutions that can be used to support complex decision making and problem solving [18]. Real-time, low latency monitoring and analysis of business events for decision making is key, but difficult to achieve [19]. The difficulties are intensified by those processes and supply chains which entail dealing with the integration of enterprise execution data across organizational boundaries. Such processes usually flow across heterogeneous systems such as business process execution language (BPEL) engines, Customer Relationship Management (CRM) systems, and Supply Chain Management (SCM) systems. The heterogeneity of these supporting systems makes the collection, integration and analysis of high volume business event data extremely difficult [20]. The new possibilities of storing and analyzing big data are changing the DSS landscape, including, for instance, decision support social networks [21].

Previous work by the authors [19] presented a big-data based DSS that provides visibility and overall business performance information on distributed processes. This DSS tool enables business users to access performance analytics data efficiently in a timely fashion, availing of performance measurements on an acceptable response time basis. This paper presents and extends a methodology presented also in [22] aimed to help users to deploy DSS tools in big data environments. In a nutshell, the aim of this method is to assist business users in sustaining a comprehensive process improvement program by means of a DSS built on Big Data. The remainder of this paper is structured as follows. Section 2 presents the five steps of the process to guide DSS implementation in Big Data environments for business process analytics. Section 3 presents a case study that depicts the application of the process in a real scenario. Finally, section 4 concludes the paper and outlines potential research directions.

## 2. Description of the process

The Business Processes Improvement (BPI) arena incorporates a plethora of methods and approaches. In spite of this fertility, BPI seems to be art rather than science [23]. To avoid getting lost in the “improvement black box” it would be

useful to have directions and rules that support the act of process improvement [24]. The effort presented in this paper is not a method for BPI. This is a method to deploy DSS in big data environments to be applied with the final aim of BPI.

Thus, the methodology presented in this paper consists of five phases (Fig. 1). The first phase identifies business process models that we aim to monitor and improve. The second phase studies and defines the physical elements of the operational systems and prepares the analytical environment for collecting enterprise performance data. This phase will identify the steps that must be undertaken within the operational environment in order to gather and collect performance information. The third phase involves the implementation of listeners for capturing and collecting both structural and behavioral information from operational systems. The fourth phase monitors the execution of processes, and establishes quality control measures in order to identify critical paths and incompliant situations. And the fifth phase leverages the outcomes obtained from the previous step to reveal deficiencies in the process that was defined in the first phase. The deficiencies found determines those processes that are susceptible to be improved. Once the improvement measures have been undertaken, the lifecycle starts over again on a continuous refinement basis.

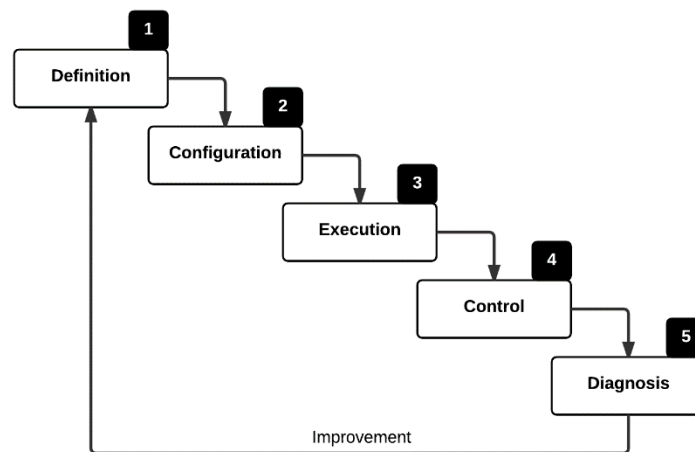


Fig. 1. A methodology for business process improvement

## 2.1 PHASE 1. Definition

In this phase, the identification of the distributed business process model along the large and complex supply chains are performed. Thus, the definition consists in discovering and defining the process that is aimed to be improved. Likewise, the purpose of this phase is not only to identify and represent the business process that has a significant value for the organization, but also to have clear insight into the strategic management of the enterprise and a good understanding of the business goals being pursued. This will help the analyst to identifying the critical processes or activities that must be monitored. For identification of the process models, authors use a method based on the tabular application development (TAD) methodology widely described in the work of Damij et al. [25]. Several steps are included in this phase, depicted as follows.

### 2.1.1 Identification of scope and boundaries

This step consists in identifying the scope and boundaries of the global business process, and defining the global business process itself. In large and complex supply chains, there are a considerable number of business entities that are involved in the business process, such as Manufacturing, Sales, Stock, Logistic, Accounting, etc. The determination of

these participants is crucial for establishing the boundaries of sub-processes and discovering key interactions between enterprises (cross-functional) or departments (inter-departmental), hereinafter business nodes.

The Fig. 2 illustrates a cross-functional business process that flows across six organizations, namely business nodes. The demand and delivery lines depict the global business process that must be identified in this step along with the business nodes involved.

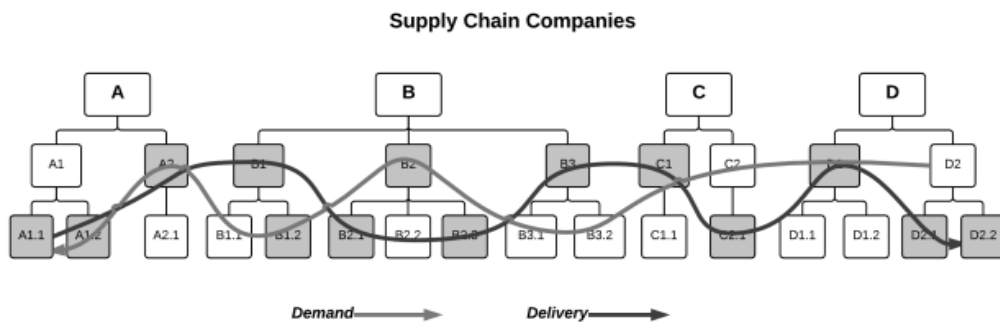


Fig. 2. Cross-functional business process

2.1.2 Definition of sub-processes, activities and sub-activities

In this step we have to iterate over each organizational node that has been identified in the previous step. For each organizational node previously defined, the aim is to discover sub-processes, activities and sub-activities (see Fig. 3) associated with the global process identified in the previous step.

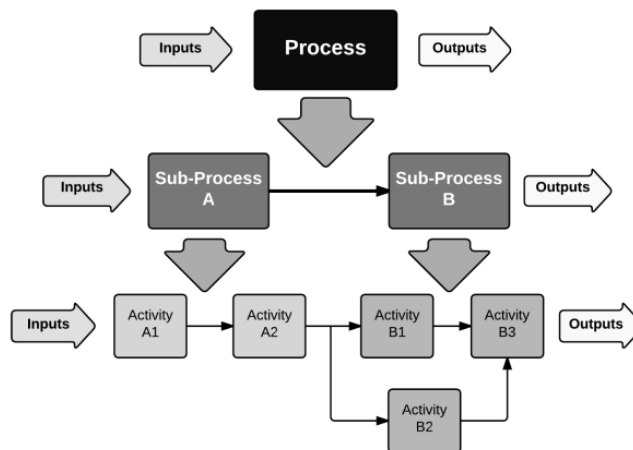


Fig. 3. Sample process hierarchy [14]

As stated, this BPI methodology is sustained by the big-data based DSS system discussed in [19]. This IT solution presents capabilities to monitor and query the structural and behavioral properties of business processes. Hence, it is required to gather properties relevant to the structure of the processes and activities. Similarly, it is imperative to focus on the input, outputs and payloads of processes and activities, as this information will be essential at further stages for establishing the link between inter-related processes.

### 2.1.3 Determination of level of detail within business processes

According to [20], as depicted in Fig. 3, “a process may itself be composed of a number of different sub-processes or activities which in turn may be decomposed into a set of smaller related tasks” [10]. There is no globally accepted limit on the number of levels, and depending on the nature of the business process and the specific requirements on process improvement endeavors, it may be necessary to monitor both high level and low level processes. The actual number of levels must be identified in this step.

The greater the number of nested levels, the more cumbersome is the deployment of the DSS, and the more complex is the monitoring and analysis of the performance information. Consequently, it is important to determine the trade-off between the deployment costs, and the final value of such data. If the performance information of an activity or sub-activities at a given level is neither crucial nor relevant, then it might be better to leave them out of the analysis. Additionally, each business node may have its own level of detail per process or activity. Every BASU (Business Analytics Service Unit) can perform the analysis of their own processes in isolation [19].

### 2.1.4 Development of model tables

The last step in this first phase is to model the business process in a tabular form. This methodology follows a business process model representation using tables because they are useful for representing the sequence of events clearly, are easy to manage for business users [25], and simplify the deployment of the DSS system in further stages.

In this step it is needed to create a table per business node (a very simplified representation of the business process model), where each table is organized as follows: the first column defines the global business process definition of the business node. Consequently, this process is a sub-process of the cross-organizational process defined in the first step. The second column presents the activities grouped by processes; the third column represents the nested level of the activity by making a reference to the parent activity. The fourth and last column lists a set of properties in the form of key value pairs.

## 2.2 PHASE 2. Configuration

During the configuration phase we prepare the analytical environment to receive structural event data from the operational systems that will feed the DSS for later analysis. Hence, this step is crucial for the overall success of the performance analysis, and equally important in the successful implementation of the DSS. During the configuration phase, software boundaries and inter-departmental processes within business nodes are identified. Likewise, the selection of the event data format, and the determination of instance correlation data are also undertaken. Finally, software listeners, along with a selection metrics and their threshold values, are established and implemented. Phase 2 consists of the steps outlined in next sections.

### 2.2.1 Business nodes provisioning and software boundaries identification

In this step the system must provision a BASU component [19] per business node identified in the Definition phase. The number of nodes may vary depending on three main factors: 1) the nature of the business process that it is intended to analyze; 2) the performance of the DSS; and 3) security issues due to the data sharing between the BASU unit and the GBAS (Global Business Analytic Service) component.

The DSS described in previous works [19] allows individual companies in a supply chain to own and manage their data. Provided data sharing was not an issue, or if a single secure data store was acceptable to all process owners, we can provide one BASU unit per business node. Otherwise, it is possible to breakdown a business node into smaller business units, and provision a unique BASU component per unit. This solution is also valid for performance reasons. Subsequently, and as part of the business nodes provisioning step, it is necessary to load the process model tables into each corresponding BASU unit.

Once we have provisioned all business nodes, we must identify the software boundaries within each business node. This will give us an insight into the software requirements on source systems when implementing the listener in a further step. Furthermore, these software boundaries are normally linked to inter-departmental sub-processes. Therefore, the use of the model tables developed in the Define phase are very useful to discover technological requirements for those processes that flow across heterogeneous systems.

### 2.2.2 Selection of event data format

The event format data that will feed the system must be established in this step. The selection of the right format will tackle the problems of integration described in [26]. According to [27], the most popular and accepted formats for process mining are XES, MXML and BPAF. The final selection of the format will depend on the business analyst and whether he or she considers it useful or not to maintain interoperability of the event logs with other process mining tools and techniques besides the DSS.

Within the DSS context, the legacy listener software may emit the event information to different endpoints depending on the message format provided. At this time, the platform presented in [19] supports a variety of widely adopted formats for representing event logs such as XES, MXML [27], [28] or even extended BPAF [28]. Every BASU unit transforms and correlates its own events by querying the event repository for previous instances. The DSS event correlation algorithm uses the event data specified in the message format, and consequently this correlation data is key for the accuracy and quality of the performance data.

### 2.2.3 Event correlation data determination

The goal of this step is the determination of which part of the message payload will be used to correlate instances. The term *instance correlation* refers to the unique identification of an event for a particular process instance or activity during execution. For instance, for an order process, the order number may be used to match the start and end of the event sequence in the timeline. Event correlation is on the critical path, and must be executed in a timely manner. Without the ability to correlate events, it is not possible to generate metrics or Key Performance Indicators (KPI) per process instance or activity [29]. Moreover, if the correlation data is not chosen in a correct way, established metrics would be incorrect, leading to a poor accuracy and loss of quality on analytical data. In this phase it is needed to look into the business process model table and identify the relationships among processes. The common properties along the business process will reveal good candidates for using their values as correlation data. Table 1 presents the identification of correlation properties.

Table 1 - Correlation properties identification on the model table

Process	Activity	Activity Parent	Properties
1#P <sub>1</sub>	1#A <sub>1</sub>		<b>Prop<sub>1</sub></b>
	2#A <sub>2</sub>	A <sub>1</sub>	<b>Prop<sub>1</sub>, Prop<sub>2</sub>, Prop<sub>3</sub></b>
	3#A <sub>3</sub>	A <sub>1</sub>	<b>Prop<sub>1</sub>, Prop<sub>2</sub></b>
2#P <sub>2</sub>	4#A <sub>4</sub>		<b>Prop<sub>1</sub></b>
	5#A <sub>5</sub>	A <sub>4</sub>	<b>Prop<sub>1</sub>, Prop<sub>4</sub></b>

### 2.2.4 Listeners implementation

In this step the software needed to collect event execution data of instances is developed. Taking into account that it must deal with the format selected in step 2, the event data must contain at least the mandatory entries stated in Table 2.

Table 2 - Event structure data.

Field	Description	Optional
<b>EventId</b>	Unique identifier for the event per business node.	
<b>Source</b>	BASU unit.	
<b>ProcessDefinitionId</b>	Definition of the process identified in model table.	
<b>ProcesName</b>	Name of the process.	X
<b>ActivityDefinitionId</b>	Definition of the activity identified in model table.	X
<b>ActivityName</b>	Name of the activity.	X
<b>ActivityParent</b>	Parent of the current sub-activity.	X
<b>StateTransition*</b>	State transition for the current event. This is highly dependent of the message format.	
<b>Correlation[]</b>	Set of key/value pairs used for correlation.	
<b>Payload[]</b>	Set of key/value pairs that represent the structural properties of the process or activity.	X

### 2.2.5 Selection of metrics and KPIs

KPIs are indispensable to build a concrete understanding of what needs to be monitored and analyzed. Within a Business Activity Monitoring (BAM) context, the construction of metrics and KPIs is intended to be performed with minimum latency, and this can be a data-intensive process in big data based DSS systems with BAM capabilities, as indicated in [19]. Hence, the metrics and KPIs must be selected carefully.

Once the metrics are triggered in the DSS, we may establish thresholds per process or activity. This decision depends whether there already exists or not in the DSS historical information where the expected execution time of a process or instance could be calculated or inferred. In such cases, the thresholds might be set in the BAM component to generate alerts, and hence detect non-compliant situations automatically. The structural metrics that the DSS is currently able to calculate are:

- **Running cases:** number of instances executed for a given process or activity;
- **Successful cases:** number of instances for a given process or activity that completed their execution successfully;
- **Failed cases:** number of instances for a given process or activity that finalized their execution with a failure state;
- **Aborted cases:** number of instances for a given process or activity that did not complete their execution.

Apart from structural metrics, the process also defines some behavioral metrics inspired by the works of [30]:

- **Turnaround:** Computes the gross execution time of a process instance or activity;
- **Wait time:** Measures the elapsed time between the entrance of a process or activity in the system and the assignment of the process or activity to a user prior to the start of its execution;
- **Change-over time:** Evaluates the elapsed time between the assignment of the process or activity to a user and the start of the execution of the process or activity;
- **Processing time:** Measures the net execution time of a process instance or activity;
- **Suspend time:** Gauges the time an execution of a process or activity is suspended.

Similarly, the methodology presented in this paper incorporates the performance dimension that is defined as a quality factor in the works of Heidari and Loucopoulos [31]. The following two measures refer to the performance dimension, and they are adapted to this methodology as KPIs that can be inferred from the metrics defined above.

CYCLE-TIME

Time is a universal and commonly used measure of performance. It is defined as the total time needed by a process or activity instance to transform a set of inputs into defined outputs [31], i.e. the total amount of time elapsed until task completion. This KPI is automatically derived from the “Turnaround” metrics defined in [30], and it is provided by the DSS.

$$T(a) = DD(a) + PD(a)$$

$a = \text{Activity.}$

$T(a) = \text{Cycle Time duration of an activity.}$

$DD(a) = \text{Delay Duration of an activity.}$

$PD(a) = \text{Process Duration of an activity (processing time).}$

$$DD(a) = CH(a) + WT(a) + ST(a)$$

$DD(a) = \text{Delay Duration of an activity.}$

$CH(a) = \text{Change over time of a process or activity.}$

$WT(a) = \text{Waiting time of a process or activity.}$

$ST(a) = \text{Suspended time of a process or activity.}$

$$OF : \text{Min}T(a)$$

$OF = \text{Objective Function.}$

TIME EFFICIENCY

This KPI is derived from the *Time Efficiency* quality factor defined in QEF. Activity *Time Efficiency* measures “how an activity execution is successful in avoiding wasted time”. This KPI is the “mean of Time Efficiency in different instances of an activity execution”. Formulae for *Time Efficiency* KPI calculation are defined as follows:

$$ET(a) = \frac{PT(a)}{T(a)} \times 100$$

$a = \text{Process or activity.}$

$ET(a) = \text{Time of Efficiency of a process or activity.}$

$T(a) = \text{Cycle time duration of a process or activity.}$

$PT(a) = \text{Planned Time duration of an activity. This is a big data based function that is inferred by the historical registry of the DSS.}$

$$OF : E(a) \geq 100$$

$OF = \text{Objective Function.}$

## 2.3 PHASE 3. Execution

During this phase the operational systems are executed making the listeners and the overall DSS fully operational. During the execution phase, the overall infrastructure is monitored including matching of defined patterns of events and real data along with expected metrics.

The next phase is only reached once the trial-execution phase is completed successfully.



#### 2.4 PHASE 4. Control

Outcomes of the overall implementation are analyzed by business users during this phase. Fig. 4 illustrates the different dimensions on which the analysis can be focused in this phase.

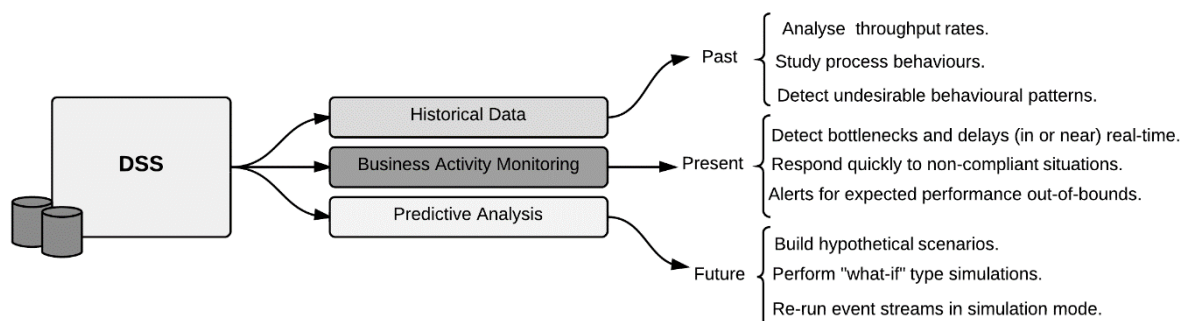


Fig. 4. Business process analytics on different dimensions

Effectively, the system processes the analytical data from three different perspectives [28]:

- 1) **Historical Analysis:** the analysis of the event logs to provide business users with a powerful understanding of what has happened in the past;
- 2) **Business Activity Monitoring:** monitoring and evaluation of what is happening at present;
- 3) **Predictive Analysis:** this will give analysts the ability to predict the behavior of process instances in the future.

#### 2.5 PHASE 5. Diagnosis

Inspired by [32], the purpose of this phase is to evaluate the improvement results and ensure whether the operation performance of the problematic processes have achieved desired results. According to [33], the improvement phase is considered to be the most creative phase during a BPM project, so personnel working in this phase must be creative and competent to extract meaningful information from results.

Thus personnel may exploit the DSS capabilities such as visualization to identify hot-spots, or re-run event streams in simulation mode in order to perform root cause analysis, among others. Once the weaknesses are found, they must be eliminated from the operational systems. In such a case, the business process is re-designed and re-deployed in the operational environment, and the improvement lifecycle starts over again on a continuous refinement basis.

### 3. Case Study

We present a case study intended to test the methodology proposed by using a big data based DSS described in [19]. The case study is focused on the improvement of the service delivery process for call centers to enhance productivity while maintaining effective customer relationships. Call centers play an essential role in the strategic operations of organizations as it directly impacts on customer loyalty and their experiences greatly influence in their decision to stay or leave that organization [34]. The provision of effective customer service is crucial for corporations in running a competitive business environment.

In our approach we model a hypothetical large-scale international company with presence in multiple countries. This fictitious enterprise provides worldwide customer service assistance. Their call centers are spread around the globe assisting customers from different regions and in multiple languages (see Fig. 7). Every call flows through one or many

call centers routing inbound calls towards the most suitable free agent to attend the request with the aim of providing the customer with the best service value. The flow of the incoming calls is modelled and represented as the target business process that we aim to monitor, analyze and improve.

Before proceeding with the description of the business process, we must first give a brief overview of how call centers internally work. Typically, a call center (see Fig. 5) is comprised of the following main components: a PABX (Private Automatic Branch Exchange); an IVR (Interactive Voice Response); multiple queue channels (normally grouped by categories); an ACD (Automatic Call Distributor); and a number of agents that handle the incoming calls. Every agent normally has a workstation that is connected to a specific-purpose enterprise information system. Usually, these systems are customer relationship management systems (CRM) or hybrid systems that complement each other to fulfil the customer demands. The PABX is the entry point to the call center and supports IVR and ACD functionality. A number of extensions are connected to a PABX, and every extension is attached to the ACD. The ACD switch is responsible for dispatching an incoming call over a certain line by selecting an extension with a free agent. An incoming call is first routed to an IVR once it succeeded to establish a communication with the trunk line in the PABX. The IVR provides standard message recording which drives the caller through a menu to select the most appropriate category to the customer. Finally, the ACD dispatches the call to the most suitable free agent. Alternatively, if the call center workload is unbalanced, inbound calls may be forwarded to another call center according to the customer requested services and needs [35].

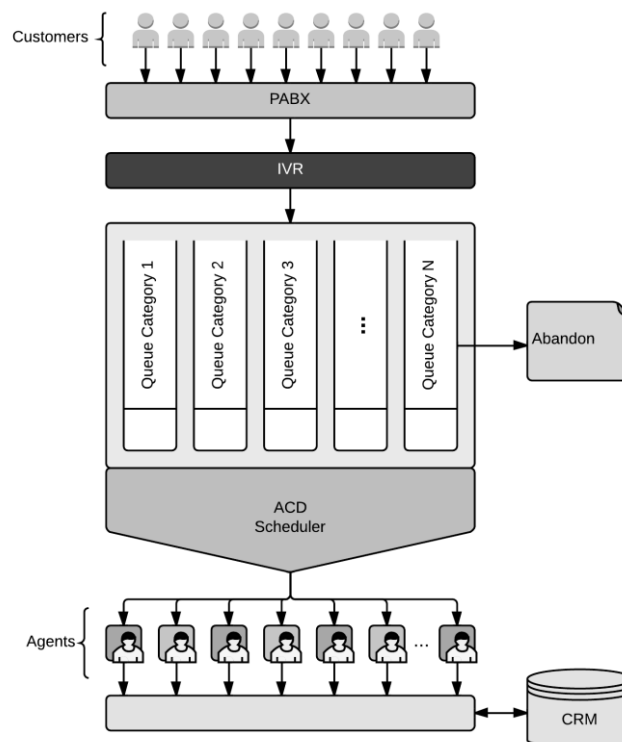


Fig. 5. Call center overview (adapted from [35]).

Based on the aforementioned assumptions, the objective business process aimed to be improved is illustrated in BPMN notation in Fig 6.

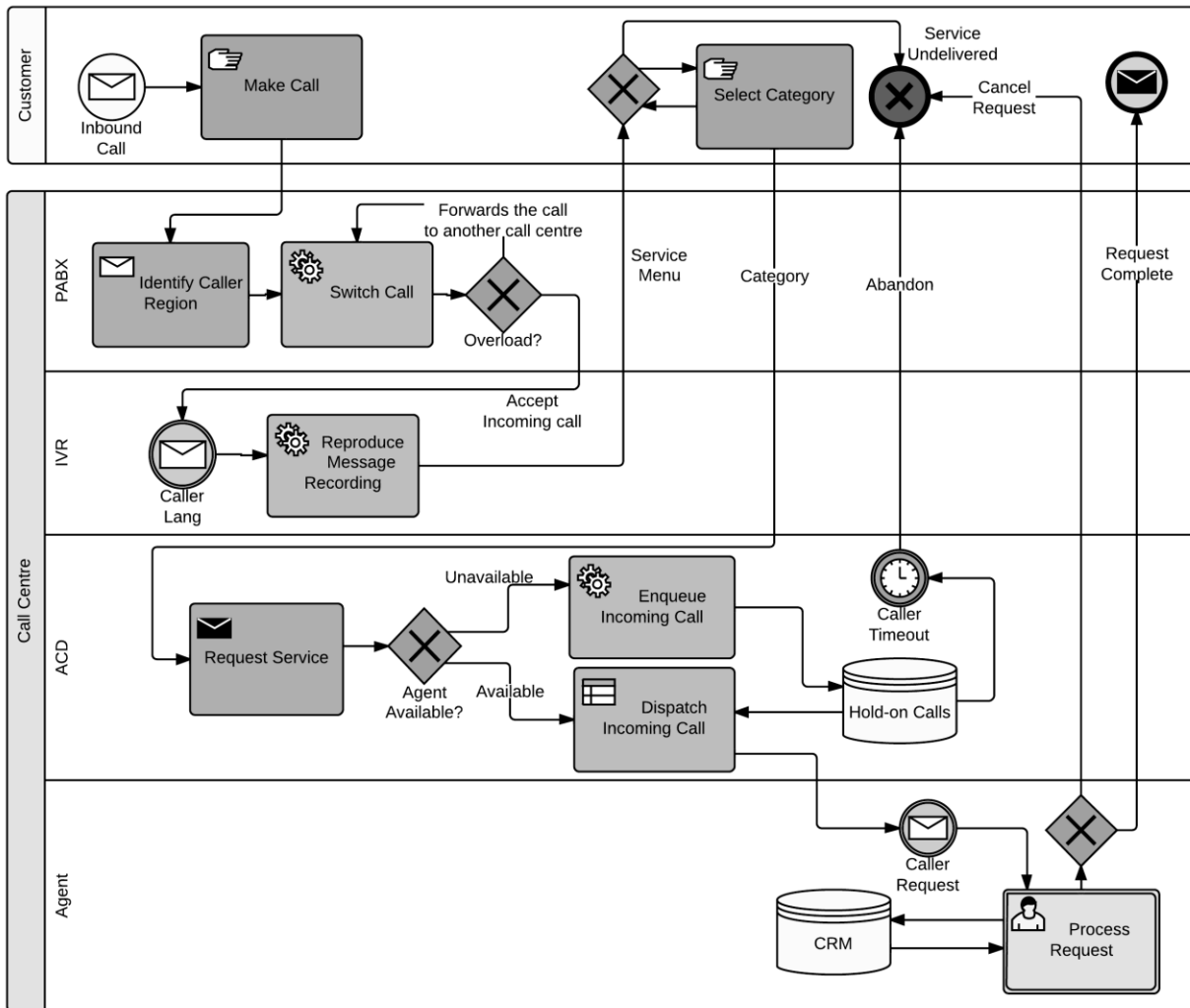


Fig. 6. Business process in BPMN notation

The improvement of the business process can drastically impact on the overall performance level of a call center. Of course, it is important not to drive the wrong type of behavior by rewarding agents for closing calls too quickly, and perhaps not dealing correctly with the customer query or problem. Notwithstanding, normal process throughput is basically measured in terms of waiting time of calls, the rates of abandons, and the productivity of the agents based on the number of calls handled and their duration. These measures will give analysts an insight into critical factors that will directly affect business process performance such as routing policies, queues distribution, overloads, abandonments, retrials, etc.

[35] outlines the following benchmarks for a well-run call center: 1) the agent underutilization level never goes up above the 5% of their total workload capacity; 2) the rate of call handling is approximately one thousand calls per hour over one hundred agents; 3) by average, a half of incoming calls are answered immediately; and 4) the abandon rate for calls on standby waiting for service ranges between a negligible 1 and 2 per cent. These high levels of service quality are very hard to accomplish for a call center, even for the most productive ones. This case study aims to achieve a global visibility of call center performance that will lay the ground for gaining an insight into the improvement of the overall quality of customer service.

The estimated volume of call arrivals is expected to be huge, whereby the number of events generated by the call center will grow considerably over time. In order to achieve a timely monitoring and analysis of call center performance, the big data based DSS system introduced in [19] has been leveraged and applied in conjunction with the proposed methodology. The implementation methodology is rolled out in the following sections.

### 3.1 PHASE 1. Definition

#### 3.1.1 Identification of scope and boundaries

In this phase we identify 18 business nodes that correspond to the different call centers that are spread around the globe. The call centers are outlined in the following table. Whereas the volume of event data tends to be huge over time, the analysis will be broken down into several distinct locations. The reason for splitting the data analysis process through many locations is twofold: 1) *performance reasons*: the vast amount of event data produced by an individual call center is easier to manage when it is stored and analyzed in isolation; and 2) *managerial reasons*: the improvement process is greatly simplified as it allows business users to perform data analysis locally on individual call centers. This enables analysts to drill down into greater level of details within the scope of a particular call center rather than dealing with a broader view of the entire business service. This is more efficient and manageable to detect and identify exceptional issues that affect the performance of other call centers. Thereby, each call center will manage its own data locally, and the data interaction between call centers will be shared among them (see Fig. 7).

Table 3 - Call center identification

Call Center ID	Location	Call Center ID	Location
CC01	USA (East)	CC10	Spain
CC02	USA (West)	CC11	Norway
CC03	Canada	CC12	Algeria
CC04	Ireland	CC13	Ukraine
CC05	Mexico	CC14	Russia
CC06	Venezuela	CC15	India
CC07	Brazil	CC16	China
CC08	Argentina	CC17	Japan
CC09	South Africa	CC18	Australia

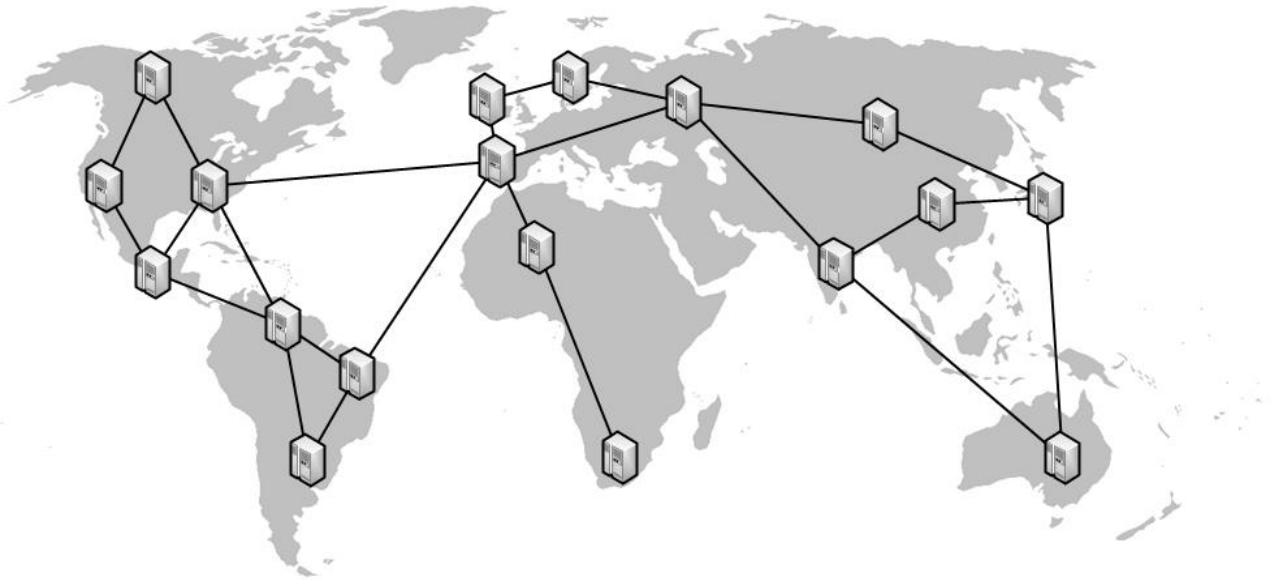


Fig. 7. Call center locations

### 3.1.2 Definition of sub-processes, activities and sub-activities

In this case study we aim to analyze the performance of the service delivery process of call centers. For this purpose, we define a global process that represents the service request that may flow through diverse call centers in different locations to cater the customer demand. Consequently, we define a sub-process as an incoming customer call that is processed within a particular call center. The activities and sub-activities of incoming calls correspond to the tasks and sub-tasks defined in the business process depicted in Fig. 6.

### 3.1.3 Determination of level of detail within business processes

The process performance improvement is intended to be performed on every call center, and this entails the monitoring and analysis of a wide range of information such as routing policies, queues distribution, overloads, abandonments, etc. Therefore, the data gathering and analysis must include the activity level of those tasks specified in the business process (see Fig. 6).

### 3.1.4 Development of model tables

For constructing the model table we have identified the activities (tasks) of the target process and determined their relevance for inclusion in the analysis. Table 4 outlines the process model developed and highlights those activities that are discarded. These tasks are rejected mainly because either they are irrelevant or supply useless information for decision making.

Table 4 - Call-center process model

Process	Activity	Activity Parent	Properties
Incoming.Call	Region.Identification*		CallID
	Swiith.Call		CallID, Country
	Message.Recording.Reproduction		CallID, Country , Category
	Request.Service		CallID, Country , Category
	Enqueue.Call		CallID, Country , Category
	Dispatch.Call		CallID, Country , Category, AgentID
	Process.Request		CallID, Country , Category, AgentID, CustomerID

*\*The "Region.Identification" activity is eliminated from the analysis because it does not affect the overall business process performance. This operation is attained by the call-center software and it is assumed it performs very quickly.*

## 3.2 PHASE 2. Configuration

### 3.2.1 Business nodes provisioning and software boundaries identification

We deployed 18 BASU nodes in a test environment for evaluating the approach. Once every business node is provisioned, the process model developed in the previous phase is loaded in every node. The BASU units deployed are outlined in Table 3. In a real case, this phase is crucial to identify the specific software requirements of every call center along with their internal information systems such as CRM, ERP, etc. The interaction among those systems gains special relevance in this step since the integration and data sharing between both will be essential when designing and implementing the listeners in a later stage. For instance, we must identify how the CallID is represented, stored and linked in the CRM system for a specific customer request. Whilst shared attributes like CallID, and CustomerID are part of the event payload, these are sourced from different systems, so this should be taken into account when implementing the listeners. In this study case, the event generation is performed using a simulation tool, and thus the analysis of the software boundaries is waived in this step.

### 3.2.2 Selection of event data format

We selected exBPAF as the event format since we do not require integration with other process mining tools. Furthermore, exBPAF does not require format conversion on the DSS since it already deals with BPAF internally.

### 3.2.3 Event correlation data determination

This phase is critical to recreate successfully the inbound customer calls across call centers. For the purpose of this case study, and assuming that the call-center software is able to generate a unique ID per call across nodes, the correlation data to be used is the identification number that is managed by call centers to identify incoming calls (CallID). This information will uniquely identify the process instance along the sequence of events.

### 3.2.4 Listeners implementation

For the implementation of the listeners we leveraged a simulation tool that generates the sequence of events according to a specific distribution function. The call duration time, volume of incoming calls, peak times, rate of abandons, and other relevant features used for diagnosis have been configured as input in the simulation engine (see PHASE 3. Execution). The aim of these specific settings is to demonstrate that the expected outputs on the DSS are those configured on the simulation side. Namely, the DSS is able to detect and identify any exceptional situation originating from the simulation. Next is a sample event generated by the listener.

```
<ns2:Event EventID="0e2c1d29-e6ea-4405-bd54-9648999d0326" ServerID="BASU-CC04"
ProcessDefinitionID="IC" ProcessName="Incoming.Call" ActivityDefinitionID="PR"
ActivityName="Process.Request" Timestamp="2014-06-25T09:14:53.190+01:00"
xmlns:ns2="http://www.uc3m.es/sofilab/basu/event">
  <EventDetails PreviousState="Open.NotRunning.Ready" CurrentState="Open.Running.InProgress"/>
  <Correlation>
    <CorrelationData>
      <CorrelationElement key="CallID" value="a7256c96-83ee-467e-bb84-135b35bbba31"/>
    </CorrelationData>
  </Correlation>
  <Payload key="Country" value="Ireland [IE]"/>
  <Payload key="Category" value="Help Desk"/>
  <Payload key="AgentID" value="27AC0491"/>
  <Payload key="CustomerID" value="19408284761"/>
</ns2:Event>
```

### 3.2.5 Selection of metrics and KPIs

The set of metrics and KPIs selected for the purpose of this case study are specified below. The DSS-standard metrics are outlined in the Table 5 for representing behavioral measures, and Table 6 for the structural ones.

Table 5 - DSS-Standard behavioral measures.

DSS-Standard Measure	Description
<b>Throughput time</b>	Total amount of time for a call to process.
<b>Change-Over time</b>	Time elapsed since a call is assigned to an agent until the agent caters the customer request.
<b>Processing time</b>	Effective amount of time for an agent to process the request.
<b>Waiting time</b>	Time elapsed for a call in on-hold state waiting for a free agent to cater the call.
<b>Suspended time</b>	Total suspension time of a call by an agent while processing the request.

Table 6 - DSS-Standard structural measures.

DSS-Standard Measure	Description
<b>Running cases</b>	Number of incoming calls processed.
<b>Successful cases</b>	Number of incoming calls that were processed successfully.
<b>Failed cases</b>	Number of incoming calls that were processed unsuccessfully (did not fulfil the customer demand).
<b>Aborted cases</b>	Number of incoming calls that abandoned the queue.

The KPI's outlined above are deduced by querying and filtering the event data gathered from the listeners. The details of how this calculation is performed are out of scope in this paper. Regarding to the KPI selection, and only for illustration purposes, we have selected the following behavioral and structural KPI's for measuring and identifying non-compliant situations (in or near) real-time.

Behavioral KPIs

*Congestion:* This KPI uses the waiting time measure and sets a threshold value for those time intervals that are susceptible to experience some congestion at peak times. This measure gives an insight into the workload of agents and the need to allocate resources during certain periods of time. This threshold value is agreed at design time during the simulation stage. When the threshold is reached, an alert is fired on the DSS.

*Agent efficiency:* This KPI measures the agent efficiency by computing the total amount of time that it takes the agent to process the customer request and the effective time used to handle the call.

$$AE(a) = Th(p_a) / Pt(P_a)$$

*AE(a) = Efficiency rate of agent "a".*

*Th(p<sub>a</sub>) = Throughput time of instances handled by agent "a" on "Process.Request" activity.*

*Pt(p<sub>a</sub>) = Processing time of instances handled by agent "a" on "Process.Request" activity.*

Structural KPIs

*Abandon rate:* This KPI computes the average rate of abandons per category. This enables the system to detect bottlenecks or inefficiencies on a determined queue or category. This is calculated by obtaining the aborted instances of the "Enqueue.Call" activity per every running instance of the "Request.Service" activity.

$$AR(c) = \frac{\forall_i \in "EnqueueCall": (AC(i): i_c = c)}{\forall_j \in "Request.Service": (RC(j): j_c = c)}$$

*AR(c) = Abandon rate KPI for the category "c".*

*AC(i) = Number of aborted cases for instances of process "Process.Request" under the category "c".*

*RC(j) = Number of running cases for instances of process "Process.Request" under the category "c".*

*Productivity:* This KPI measures the productivity of the call center. This is calculated by obtaining the successful instances of the "Process.Request" activity for every running instance of the "Request.Service" category.

$$P = \frac{\forall_i \in "Process.Request": SC(i)}{\forall_j \in "Request.Service": RC(j)}$$

*P = Productivity of the call center.*

*AC(i) = Number of successful cases for instances of process "Process.Request".*

*RC(j) = Number of running cases for instances of process "Request.Service".*



*Overload:* This KPI measures the number of correlated events across call centers. The objective function counts the number of executions of the “Switch.Call” activity. When a call center is overloaded the software switches the call to an alternative node, thereby generating a new activity on the target call center with the same CallID but with different source.

### 3.3 PHASE 3. Execution

The evaluation has been accomplished successfully in a test environment that follows the infrastructure depicted on the Fig. 8. A large amount of event data was generated by the simulation tool whereby inbound calls were generated in order to simulate flows that cross multiple call centers. Moreover, different scenarios were built and configured in the simulation engine in order to produce the desired outcomes on the DSS. These hypothetical cases aimed to detect exceptional situations such as overload, low running resources on peak times, high abandon rates, etc.

The simulation model was based on a discrete event simulation approach. The simulation was built using DESMO-J, which is a java-based simulation library that supports both event-oriented and process-oriented modelling approaches. The events generated from the simulation model were persisted before being forwarded to the specific event channels for processing on the DSS side. The model implementation used three main entity types:

- Calls: whose properties stored details about the caller ID, caller location, calling time and service category;
- Call agents: which hold references to the call center in which they are located and which type of service that each agent can help with;
- Call centers: which store information about the call centers locations and the backup centers in case of unbalancing.

The model defined six different classes of events, the Table 7 presents the events and their descriptions.

Table 7 – List of events and descriptions.

Event	Description
<b>Incoming Call</b>	A new call arrival at a defined point of time.
<b>Dispatch Call</b>	An idle agent is assigned to handle an incoming or awaiting call.
<b>Service End</b>	A call was successfully handled by a call agent.
<b>Enqueue Call</b>	A call was put on-hold because all agents are busy.
<b>Switch Call</b>	A call has been switched to another call center in case that the max on-hold time was exceeded.
<b>Abandon Queue</b>	A call abandoned the queue.

The model included four queues of idle call agents in each call center, where each queue represents a different category of service. Similarly, each call center had four queues of awaiting calls. Since the simulation scenario involved 18 call centers in different locations, 144 queues, collectively, were needed to be created during each simulation experiment. In addition, the event listeners were represented as *dispatchers*. The dispatcher is a core component responsible for relaying the events generated from the simulation engine to the DSS. The dispatcher included the capability to control the timing of the transmitted messages, which could be used to measure the capacity of the framework.

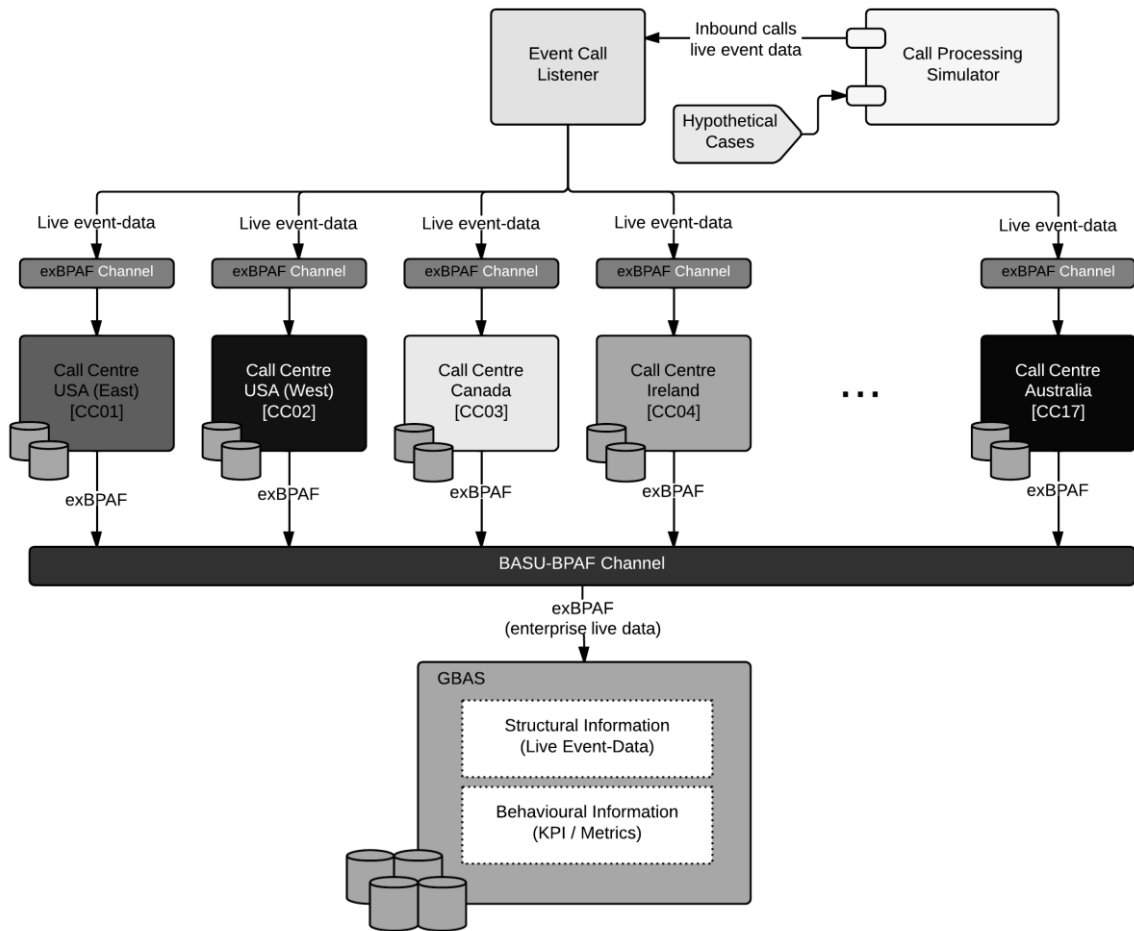


Fig. 8. DSS infrastructure

3.4 PHASE 4. Control

We successfully experienced that the outcomes of the DSS were those expected. The execution outcomes, measures and KPIs did not present any statistical significance in respect with the values set in the simulation engine as input. Likewise, exceptional cases such as bottlenecks, overloads and failure rates (abandons) were properly identified and detected by the system.

3.5 PHASE 5. Diagnosis

This phase is out of scope in this paper since we are designing a case study based on a simulated environment through the use of models that represent diverse hypothetical cases.

#### 4. Conclusions and future work

This paper has presented a methodology and system which leverages the scalability and processing power of Big Data to provide business process monitoring and analysis across complex, multi-level supply chains. The system itself is extensible, and allows a number of event formats to be used in the data collection. The case study has demonstrated the functionality and robustness of the implementation. By using a simulation to generate event data in any quantity desired, and running it in either real-time or in accelerated mode, we can test the scalability of the system. Further work will be devoted to applying the methodology and framework to a variety of application domains, such as manufacturing, logistics and healthcare. Each domain has its own interfacing issues, process and organizational configurations, as well specialized performance measurements. For example, this approach should be highly useful in a distributed, decentralized "system of systems" such as healthcare, where individual business units need their own performance monitoring and evaluation. At the same time, the national health services need to monitor and improve efficiencies and outcomes along multiple care pathways. Additional work is also need to develop improved data visualization and 'playback' facilities for the system to allow process engineers to view and drill-down into aggregate and individual event data.

#### References

- [1] F. J. García Peñalvo, R. Colomo Palacios, and J. Y.-J. Hsu, "Discovering knowledge through highly interactive information based systems," *Journal of Information Science and Engineering*, vol. 29, no. 1, 2013.
- [2] R. H. L. Chiang, P. Goes, and E. A. Stohr, "Business Intelligence and Analytics Education, and Program Development: A Unique Opportunity for the Information Systems Discipline," *ACM Transactions on Management Information Systems*, vol. 3, no. 3, pp. 12:1–12:13, Oct. 2012.
- [3] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, 2014.
- [4] T. Costello and B. Prohaska, "2013 Trends and Strategies," *IT Professional*, vol. 15, no. 1, pp. 64–64, 2013.
- [5] V. R. Borkar, M. J. Carey, and C. Li, "Big data platforms: What's next?," *XRDS*, vol. 19, no. 1, pp. 44–49, Sep. 2012.
- [6] T. Kraska, "Finding the Needle in the Big Data Systems Haystack," *IEEE Internet Computing*, vol. 17, no. 1, pp. 84–86, 2013.
- [7] D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker, "Interactions with big data analytics," *interactions*, vol. 19, no. 3, pp. 50–59, May 2012.
- [8] J. Torres-Niño, A. Rodríguez-González, R. Colomo-Palacios, E. Jiménez-Domingo, and G. Alor-Hernandez, "Improving Accuracy of Decision Trees Using Clustering Techniques," *Journal of Universal Computer Science*, vol. 19, no. 4, pp. 483–500, Feb. 2013.
- [9] R. Colomo-Palacios, L. López-Cuadrado, I. González-Carrasco, and J. García-Peñalvo, "SABUMO-dTest: Design and evaluation of an intelligent collaborative distributed testing framework," *Computer Science and Information Systems*, vol. 11, no. 1, pp. 29–45, 2014.
- [10] N. Piedra, E. Tovar, R. Colomo-Palacios, J. Lopez-Vargas, and J. A. Chicaiza, "Consuming and producing linked open data: the case of OpenCourseWare," *Program: electronic library and information systems*, vol. 48, no. 1, pp. 16–40, Jan. 2014.
- [11] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz, "Big data, analytics and the path from insights to value," *MIT Sloan Management Review*, vol. 51, no. 2, pp. 21–32, 2011.

- [12] R. Sharma, S. Mithas, and A. Kankanhalli, "Transforming decision-making processes: a research agenda for understanding the impact of business analytics on organisations," *European Journal of Information Systems*, vol. 23, no. 4, pp. 433–441, Jul. 2014.
- [13] G.-H. Kim, S. Trimi, and J.-H. Chung, "Big-data Applications in the Government Sector," *Communications of the ACM*, vol. 57, no. 3, pp. 78–85, Mar. 2014.
- [14] B. Wixom, T. Ariyachandra, D. Douglas, M. Goul, B. Gupta, L. Iyer, U. Kulkarni, J. Mooney, G. Phillips-Wren, and O. Turetken, "The Current State of Business Intelligence in Academia: The Arrival of Big Data," *Communications of the Association for Information Systems*, vol. 34, Article 1, Jan. 2014.
- [15] D. I. Sessler, "Big Data – and its contributions to peri-operative medicine," *Anaesthesia*, vol. 69, no. 2, pp. 100–105, Feb. 2014.
- [16] J. L. Schnase, D. Q. Duffy, G. S. Tamkin, D. Nadeau, J. H. Thompson, C. M. Grieg, M. A. McInerney, and W. P. Webster, "MERRA Analytic Services: Meeting the Big Data challenges of climate science through cloud-enabled Climate Analytics-as-a-Service," *Computers, Environment and Urban Systems*, vol. In press.
- [17] R. Dutta, A. Morshed, J. Aryal, C. D'Este, and A. Das, "Development of an intelligent environmental knowledge system for sustainable agricultural decision support," *Environmental Modelling & Software*, vol. 52, pp. 264–272, Feb. 2014.
- [18] J. P. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson, "Past, present, and future of decision support technology," *Decision Support Systems*, vol. 33, no. 2, pp. 111–126, Jun. 2002.
- [19] A. Vera-Baquero, R. Colomo-Palacios, and O. Molloy, "Business Process Analytics Using a Big Data Approach," *IT Professional*, vol. 15, no. 6, pp. 29–35, 2013.
- [20] A. Vera-Baquero and O. Molloy, "A Framework to Support Business Process Analytics," in *Proceedings of the International Conference on Knowledge Management and Information Sharing*, Barcelona, Spain, 2012, pp. 321–332.
- [21] F. Antunes and J. P. Costa, "Reviewing Motivations for Engaging in Decision Support Social Networks:," *International Journal of Human Capital and Information Technology Professionals*, vol. 5, no. 1, pp. 1–14, 2014.
- [22] A. Vera-Baquero, R. Colomo-Palacios, and O. Molloy, "Towards a Process to Guide Big Data Based Decision Support Systems for Business Processes," *Procedia Technology*, vol. 16, pp. 11–21, 2014.
- [23] J. M. Hall and M. E. Johnson, "When should a process be art, not science?," *Harvard business review*, vol. 87, no. 3, pp. 58–65, 2009.
- [24] G. Zellner, "A structured evaluation of business process improvement approaches," *Business Process Management Journal*, vol. 17, no. 2, pp. 203–237, Apr. 2011.
- [25] N. Damij, T. Damij, J. Grad, and F. Jelenc, "A methodology for business process improvement and IS development," *Information and Software Technology*, vol. 50, no. 11, pp. 1127–1141, Oct. 2008.
- [26] G. Bruno, "A Notation for the Task-Oriented Modeling of Business Processes:," *International Journal of Human Capital and Information Technology Professionals*, vol. 3, no. 3, pp. 42–53, 2012.
- [27] J. Becker, M. Matzner, O. Müller, and M. Walter, "A Review of Event Formats as Enablers of Event-Driven BPM," in *Business Process Management Workshops*, vol. 99, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Springer Berlin Heidelberg, 2012, pp. 433–445.
- [28] M. zur Muehlen and K. D. Swenson, "BPAF: A Standard for the Interchange of Process Analytics Data," in *Business Process Management Workshops*, M. zur Muehlen and J. Su, Eds. Springer Berlin Heidelberg, 2011, pp. 170–181.

- [29] O. Molloy and C. Sheridan, "A Framework for the use of Business Activity Monitoring in Process Improvement," *E-Strategies for Resource Management Systems: Planning and Implementation*. IGI-Global, USA, 2010.
- [30] M. zur Mühlen and R. Shapiro, "Business Process Analytics," in *Handbook on Business Process Management 2*, J. vom Brocke and M. Rosemann, Eds. Springer Berlin Heidelberg, 2010, pp. 137–157.
- [31] F. Heidari and P. Loucopoulos, "Quality evaluation framework (QEF): Modeling and evaluating quality of business processes," *International Journal of Accounting Information Systems*, vol. 15, no. 3, pp. 193–223, Sep. 2014.
- [32] K. T. Lee and K. B. Chuah, "A SUPER methodology for business process improvement - An industrial case study in Hong Kong/China," *International Journal of Operations & Production Management*, vol. 21, no. 5/6, pp. 687–706, May 2001.
- [33] T. Falk, P. Griesberger, F. Johannsen, and S. Leist, "Patterns For Business Process Improvement-A First Approach," presented at the 21st European Conference on Information Systems, Utrecht, Belgium, 2013, p. Paper 151.
- [34] P. Taylor, C. Baldry, P. Bain, and V. Ellis, "'A Unique Working Environment': Health, Sickness and Absence Management in UK Call Centres," *Work, Employment & Society*, vol. 17, no. 3, pp. 435–458, Sep. 2003.
- [35] R. Doomun and N. V. Jungum, "Business process modelling, simulation and reengineering: call centres," *Business Process Management Journal*, vol. 14, no. 6, pp. 838–848, 2008.

**Biographical notes****Alejandro Vera-Baquero**

Alejandro Vera-Baquero is a PhD candidate at the Universidad Carlos III de Madrid. He is also Senior Software Engineer at the Research & Development department in Synchronoss Technologies Ireland. He previously worked as IT Consultant and Project Manager in several firms within the software industry and has more than ten years of experience leading software projects. He also has professional experience as entrepreneur where he gained and led to success the procurement of several software development contracts for the public administration. His research interests include business process improvement, big data, and business analytics. Vera-Baquero holds a Bachelor of Science in Computer Engineering from the Universidad de La Laguna and received his MSc in software engineering and database technologies from National University of Ireland, Galway.

[www.shortbio.net/averabaq@gmail.com](http://www.shortbio.net/averabaq@gmail.com)

**Ricardo Colomo-Palacios**

Full Professor at the Computer Science Department of the Østfold University College, Norway. Formerly he worked at Universidad Carlos III de Madrid, Spain. His research interests include applied research in Information Systems, IT project management and people in IT projects among others. He received his PhD in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as Software Engineer, Project Manager and Software Engineering Consultant in several companies including Spanish IT leader INDRA. He is also an Editorial Board Member and Associate Editor for several international journals and conferences and Editor in Chief of International Journal of Human Capital and Information Technology Professionals. He has published more than two hundred works in journals, books and conferences.

[www.shortbio.net/ricardo.colomo-palacios@hiof.no](http://www.shortbio.net/ricardo.colomo-palacios@hiof.no)

**Owen Molloy**

Owen Molloy is a lecturer in information technology at the National University of Ireland, Galway. His research interests include enterprise computing, intelligent enterprise integration, and business performance management. Molloy received his PhD in industrial engineering from National University of Ireland, Galway. He previously worked with Bäurer Software and Consulting GmbH. Prior to that he worked as a Senior Software Engineer for Nortel Networks, where he specialized in the implementation of Enterprise Information Systems, to improve process visibility and information flow across the supply chain. He is an active reviewer and evaluator for the European Commission since 2002, primarily in the areas of ICT, SME's and Marie Curie. Dr. Molloy has experience as a project manager in both nationally funded and EU-funded research projects.

[www.shortbio.net/owen.molloy@nuigalway.ie](http://www.shortbio.net/owen.molloy@nuigalway.ie)

**Mahmoud Elbattah**

Mahmoud Elbattah is a Doctoral student at the National University of Ireland, Galway. He is a holder of a College of Engineering & Informatics Scholarship. Elbattah holds a BSc in Computer Science, and received his MSc in Computer Science from the Arab Academy for Science, Technology and Maritime Transportation, Egypt, for work on a Fuzzy-Based Framework for ERP System Selection. He is current pursuing research in the area of Supply Chains Modelling and Simulation, investigating a Graph-Driven Approach Using Ontology-Based Semantic Networks and Graph Databases.

[www.shortbio.net/m.elbattah1@nuigalway.ie](http://www.shortbio.net/m.elbattah1@nuigalway.ie)